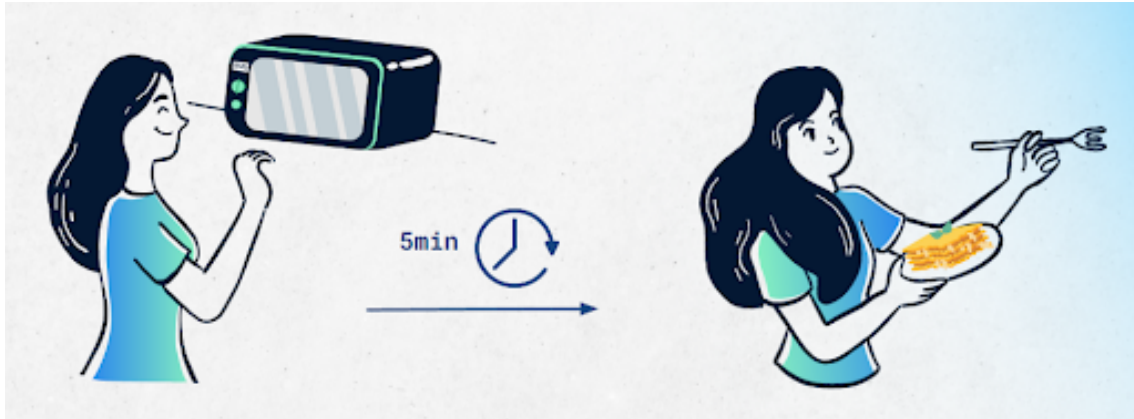


A Mônica estava com fome e colocou uma lasanha no microondas por 5 minutos. Após esse tempo, ela comerá a lasanha no almoço. Esse processo de espera é algo que podemos visualizar no javascript com os **callbacks**.



Selecione as alternativas que descrevem a utilidade dos callbacks:



Geralmente, callbacks são executados quando alguma operação é concluída ou quando um evento específico ocorre.



Callbacks são assíncronos, portanto são funções que são ativadas por algum fator pré-determinado, podendo ser um tempo específico, a partir de uma ação do usuário, depois da conclusão de alguma coisa.

**B**

Callbacks são sempre executados antes de qualquer função, ou seja, independente do lugar que ele estiver no código, será priorizado.



O javascript, através do hoisting e não de callbacks, prioriza a leitura de variáveis e declarações de funções. Apesar disso, ele não prioriza a execução das funções.

**C**

Callbacks são funções síncronas, que são executadas no momento em que o event loop a detecta.



Callbacks fazem parte da construção de funções assíncronas, pois são funções que são declaradas em um local mas que serão executadas futuramente em decorrência de algum fator.



O callback é uma função que é passada como argumento para uma outra função.



Podemos chamar de callback funções que são passadas como parâmetro para outras funções, como por exemplo a função `mandaMensagem()` que é enviada como parâmetro para a função `setTimeout`, dessa maneira: `setTimeout(mandaMensagem, 5000)`.